WILLIAM L. ANTHONY (State Bar No. 106908)
ERIC L. WESENBERG (State Bar No. 139696)
HEIDI L. KEEFE (State Bar No 178960)
KENNETH J. HALPERN (State Bar No. 187663)
SAM O'ROURKE (State Bar No. 205233)
ORRICK, HERRINGTON & SUTCLIFFE, LLP
1000 Marsh Road
Menlo Park, CA 94025
Telephone: (650) 614-7400
Facsimile: (650) 614-7401

STEVEN ALEXANDER (admitted Pro Hac Vice)
KRISTIN L. CLEVELAND (admitted Pro Hac Vice)
JAMES E. GERINGER (admitted Pro Hac Vice)
JOHN D. VANDENBERG
KLARQUIST SPARKMAN, LLP
One World Trade Center, Suite 1600
121 S.W. Salmon Street
Portland, OR 97204
Telephone: (503) 226-7391
Facsimile: (503) 228-9446

Attorneys for Defendant and Counterclaimant,
MICROSOFT CORPORATION

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

OAKLAND DIVISION

INTERTRUST TECHNOLOGIES CORPORATION, a Delaware corporation,

Plaintiff,

v.

MICROSOFT CORPORATION, a Washington corporation,

Defendant.

AND RELATED CROSS-ACTION.

CASE NO. C01-1640 SBA (MEJ)

Consolidated with C 02-0647 SBA (MEJ)

MICROSOFT'S NOTICE OF MOTION AND MEMORANDUM IN SUPPORT OF MOTION FOR PARTIAL SUMMARY JUDGMENT OF INVALIDITY OF THE ASSERTED CLAIMS OF THE '900 PATENT (ANTICIPATION)

Date: March 30, 2004
Time: 1:00 p.m.
Judge: Saundra B. Armstrong

TABLE OF CONTENTS

TABLE OF AUTHORITIES

CASES

STATUTES

NOTICE OF MOTION

Pursuant to Fed. R. Civ. P. 56(b) and 35 U.S.C. § 102(b), Defendant Microsoft Corporation ("Microsoft") respectfully moves for Partial Summary Judgment of Invalidity of the Asserted Claims of U.S. Patent No. 5,892,900. This motion is noticed for March 30, 2004 at 1:00 p.m. and is based upon this Notice and Memorandum of Points and Authorities, the Declaration of Eric Wesenberg and exhibits thereto. Pursuant to the Court's Standing Order, Microsoft met and conferred with counsel for InterTrust prior to filing this motion. Declaration of Eric L. Wesenberg in Support of Microsoft's Motion for Partial Summary Judgment of Invalidity of the Asserted Claims of the '900 Patent at ¶ 6.

MEMORANDUM OF POINTS AND AUTHORITIES

I. INTRODUCTION

Microsoft moves for summary judgment of invalidity of claims 155, 156, and 157 of U.S. Patent No. 5,892,900 ("the '900 Patent"), pursuant to 35 U.S.C. § 102(b), based on the anticipatory disclosure of the prior art U.S. Patent No. 5,113,518 ("the Durst Patent" or "the Durst reference"). The Durst Patent issued more than one year prior to August 12, 1996, the priority date InterTrust claims for the '900 Patent, and discloses every limitation of claims 155, 156 and 157 of that patent. InterTrust did not cite Durst during the prosecution of the '900 Patent and, therefore, the examiner did not take it into consideration in examining the claims that are challenged herein. Granting this motion will render claims 155, 156 and 157 of the '900 Patent invalid, simplifying this case by disposing of that patent altogether (these are the only '900 claims asserted by InterTrust). This will eliminate the need for the jury to learn and understand (i) the '900 Patent as a whole, (ii) the machine signature programming these claims represent; (iii) the details of over 100 infringement arguments that are unique to these claims, and (iv) product activation technology altogether, as there would be no claims asserted against such product activation technology remaining in the case.

II. LEGAL STANDARD

A. Legal Standard For Summary Judgment

The Federal Circuit has repeatedly emphasized that "[s]ummary judgment is as

appropriate in a patent case as in any other." *See Avia Group International, Inc. v. L.A. Gear California, Inc.*, 853 F.2d 1557, 1561 (Fed. Cir. 1988); *Spectra Corp. v. Lutz*, 839 F.2d 1579, 1581 n. 6, (Fed. Cir. 1988); *Brenner v. United States*, 773 F.2d 306, 307 (Fed. Cir. 1985). "Where no genuine issue of material fact remains and the movant is entitled to judgment as a matter of law, the court should utilize the salutary procedure of Fed. R. Civ. P. 56 to avoid unnecessary expense to the parties and wasteful utilization of the jury process and judicial resources." *Barmag Barmer Maschinenfabrik AG v. Murata Machinery, Ltd.*, 731 F.2d 831, 835 (Fed. Cir. 1984); *Brassica Protection Products LLC v. Sunrise Farms (In re Cruciferous Sprout Litig.*, 301 F.3d 1343, 1346 (Fed. Cir. 2002) ("Summary judgment is appropriate when there is no genuine issue of material fact and the moving party is entitled to judgment as a matter of law.").

Summary judgment is warranted when the moving party has demonstrated that there is no genuine issue as to any material fact and the moving party is entitled to a judgment as a matter of law. *See* Fed. R. Civ. P. 56(c). A fact is material if it

> "might affect the outcome of the suit under the governing law." *Anderson v. Liberty Lobby, Inc.*, 477 U.S. 242, 248 (1986). "With respect to whether there is a genuine issue, the court may not simply accept a party's statement that a fact is challenged. (Citations omitted). The party opposing the motion must point to an evidentiary conflict created on the record at least by a counter statement of a fact or facts set forth in detail in an affidavit by a knowledgeable affiant. Mere denials or conclusory statements are insufficient."

Barmag, 731 F.2d at 835-36.

B. Legal Standard For Patent Invalidity

1. Requirements of 35 U.S.C. § 102(b)

A party challenging the validity of a patent claim has the burden of showing invalidity by clear and convincing evidence. *Brassica*, 301 F.3d 1343, 1349 (Fed. Cir. 2002). Microsoft moves for summary judgment of invalidity based on 35 U.S.C. § 102(b), which states that an individual is not entitled to a patent if their claimed invention "was patented or described in a printed publication in this or a foreign country ... more than one year prior to the date of the application for patent in the United States." 35 U.S.C. § 102(b). Summary judgment should be granted where the defendant demonstrates that each element of the challenged claim is disclosed

in a single prior art reference. *See id.*; *Brown v. 3M*, 265 F.3d 1349, 1351 (Fed. Cir. 2001).

The Durst Patent was filed on June 3, 1988 and issued on May 12, 1992. InterTrust claims a priority date of August 12, 1996 for the '900 Patent. The Durst Patent issued more than four years before the purported effective filing date of the '900 Patent and thus indisputably is prior art to the '900 Patent. Also, as will be shown below, its specification discloses all elements of claims 155, 156 and 157 of the '900 Patent. The Durst reference is therefore invalidating prior art under 35 U.S.C. § 102(b), as the purported invention of claims 155-157 "was ... described in a printed publication in this ... country ... more than one year prior to the date of the application for patent in the United States" for the '900 Patent.

2. Presumption of Enablement

In addition to preceding the challenged patent claims by more than one year and disclosing all of the claim elements, an anticipatory reference must enable one of skill in the art to reduce the disclosed invention to practice. *Amgen Inc. v. Hoechst Marion Roussel, Inc.*, 314 F.3d 1313, 1354 (Fed. Cir. 2003). As an issued U.S. patent, the Durst reference carries a presumption that it is enabling, even as to the unclaimed material in its disclosure. *Id.* at 1355 ("We hold that an accused infringer should be ... entitled to have the district court presume the enablement of unclaimed (and claimed) material in a prior art patent defendant asserts against a plaintiff"). It is InterTrust's burden to overcome the presumption of enablement by bringing forward evidence of non-enablement. *Id.*

III. ARGUMENT

A. Overview of the Challenged Claims and the Durst Patent

1. Claims 155, 156 and 157 of the '900 Patent

Claims 155, 156 and 157 of the '900 Patent each claim the same device, differing from each other only with regard to the final element:

	Claim Language
	A virtual distribution environment comprising a first host processing environment comprising
(hardware)	a central processing unit; main memory operatively connected to said central processing unit;

	mass storage operatively connected to said central processing unit and said main memory; said mass storage storing tamper resistant software designed to be loaded into said main memory and executed by said central processing unit,
(software)	said tamper resistant software comprising: machine check programming which derives information from one or more aspects of said host processing environment, one or more storage locations storing said information; integrity programming which causes said machine check programming to derive said information, compares said information to information previously stored in said one or more storage locations, and generates an indication based on the result of said comparison; and programming which takes one or more actions based on the state of said indication; said one or more actions including ...
Claim 155	... at least temporarily halting further processing.
Claim 156	... at least temporarily disabling certain functions.
Claim 157	... displaying a message to the user.

The claimed device consists of a virtual distribution environment ("VDE") made up of a host processing environment ("HPE") comprising standard personal computer hardware – a central processing unit ("CPU"), main memory (*e.g.*, RAM) and mass storage (*e.g.*, disk drive) – operationally connected to each other so that each can perform its familiar function. The mass storage stores software capable of being loaded into main memory and executed by the CPU.

The claimed software has three aspects: (i) machine check programming, which derives information from one or more aspects of the HPE and stores it in one more storage locations; (ii) integrity programming, which activates the machine check programming to derive the same information and compares it to the information previously stored, and (iii) programming that takes one or more actions depending on the result of the comparison. As will be shown below, the claim elements make out a programming structure that the Durst reference disclosed more than four years before the '900 Patent application was filed.

Before engaging in an element-by-element comparison, it is useful to look at the claims as a whole. The specification of the '900 Patent provides context and sheds light on the purpose and function of the claimed purported invention. Programming that derives information

about a system, compares it to previously stored, similar information, and takes protective action based on that comparison is well-known in the art – the derived, stored information is often called a "machine signature." The '900 specification contains a discussion of machine signatures that discloses program features corresponding to those of claims 155-57.

The disclosed "machine signature" technique involves two programming modules: the "installation materials" and the "operational materials":

> The installation materials **3470** may be executed by computer **3372** to install the operational materials **3472** onto the computer's hard disk **3376**. The computer **3372** may then execute the operational materials **3472** from its hard disk **3376** to provide software-based protected processing environment **650** and associated software-based tamper resistant barrier **672**.

'900 Patent, 231:25-31.

The installation materials derive a machine signature from the electronic appliance and embed that signature into the operational materials. Then, when the operational materials are initialized on an appliance, they derive the machine signature of the appliance and compare it to the embedded signature:

> Correspondence Between Installed Software and Appliance "Signature".
>
> Another technique that may be used during the installation routine **3470** is to customize the operational materials **3472** by embedding a "machine signature" into the operational materials to establish a correspondence between the installed software on a particular electronic appliance **600** (FIG. **69C**, block **3470(7)**). This technique prevents a software-based PPE **650** from being transferred from one electronic appliance **600** to another (except through the use of the appropriate secure, verified backup mechanism).
>
> For electronic appliances **600** where it is feasible to do so, the installation procedure **3470** may determine unique information about the electronic appliance **600** (*e.g.*, a "signature" SIG in the sense of a unique value--not necessarily a "digital signature" in the cryptographic sense). Installation routine **3470** embeds the electronic appliance "signature" SIG in the installed operational materials **3472**. Upon initialization, the operational materials **3472** validate the embedded signature value against the actual electronic appliance **600** signature SIG, and may refuse to start if the comparison fails.

'900 Patent, 239:4-25. This language is followed by a description of how various machine parameters can be used to generate signatures. *Id.*, 239:26-240:42. To summarize, the

installation programming embeds a machine signature in the "PPE" ("Protected Processing Environment") software, which embedded signature is validated each time the PPE is initialized by comparing it to the machine signature of the current machine. If the two signatures do not match, reflecting that the PPE software has been transferred to a different, unauthorized machine, the PPE refuses to start.

2. The Durst Reference – Overview

The Durst Patent, titled "Method and System for Preventing Unauthorized Use of Software," discloses the same arrangement, functioning in the same manner, with the same elements. The Durst system also has the same purpose as the claimed '900 Patent's system – to prevent the use of software on an unauthorized computer. The abstract of the Durst Patent succinctly captures its close similarity to the apparatus in claims 155-157 of the '900 Patent:

> A technique is disclosed for preventing a computer program from being used by a computer system other than a designated system. The values of certain characteristics exhibited by the designated computer system first are stored, and then the values of those same characteristics exhibited by the computer system which is intended to use the computer program are measured and compared to the stored values. If the compared values are substantially the same, the computer program may be executed. However, if they are different, the computer system which was intended to use the program is inhibited from executing that program.

And, just as in the '900 Patent, Durst discloses embedding the machine signature in the software itself. Durst, 26:14-21; 27:11-13. The sections that follow show in detail that Durst discloses each and every element of these three '900 Patent claims.

3. The System Environment

The three '900 Patent claims first recite the computing context in which the programming operates. These basic elements are as follows:

Claim Language	A virtual distribution environment comprising

As construed by the Court, this element is simply the sum of the other elements that follow. A "virtual distribution environment" is "defined by the elements of 900.155 [claim 155 of the '900 Patent]; it has no definition independent of those elements." Order Denying

Motion for Partial Summary Judgment and Construing "Mini-*Markman* Claims" ("Markman Order"), July 3, 2003, at 55.[1] Therefore, the Durst reference need not disclose it as such.[2]

Claim Language	**a first host processing environment comprising**

The Court has defined "host processing environment" ("HPE") to mean "capabilities available to a program running on a computer or other device or to the user of a computer or other device," which, "[d]epending on the context ... may be in a single device (*e.g.*, a personal computer) or may be spread among multiple devices (*e.g.*, a network)." Markman Order, at 45. There is a further distinction between a non-secure HPE and a secure HPE, the latter having two additional features: its "processing and/or data is at least in part protected from tampering," and it incorporates "software-based security." *Id.*

The Durst reference discloses "HPEs" of both types. First, the Durst reference discloses that its technology is to be used within a computer system. Durst, Fig. 1, and 5:60-64. Second, the software is "tamper-resistant" ("make[s] tampering more difficult and/or allow[s] detection of tampering," Markman Order, at 51). Durst discloses an embodiment in which the machine signature is itself stored within the software in encrypted form and can thereafter be altered only with a password provided by the manufacturer. In this embodiment, the manufacturer will first confirm that the customer has modified the system hardware and is authorized to receive a new password. Durst, 26:14-21; 27:11-13; 28:6-27. Additionally, the software may be programmed to change the encrypted key after re-recording the machine signature so that each password may be used only once. Durst, 28:3-27. The encryption makes it more difficult to tamper with the machine signature, which is both part of the software's code and central to its authorization functions.

///

---

[1] The same would presumably apply to the VDE element of claims 156 and 157, which employ the term "VDE" in exactly the same fashion as claim 155 and which are otherwise almost identical to claim 155.

[2] Microsoft maintains its argument that "VDE" is the "present invention" identified in the '900 Patent ('900 Patent, 2:19-32), and that the asserted claims are invalid for lack of written description (35 U.S.C. § 112), non-enablement and are not infringed.

Finally, the system described in Durst incorporates "software-based security." The Court has construed "secure" to mean employing "[o]ne or more mechanisms ... that prevent or discourage ... misuse of or interference with information or processes for the purpose of discouraging and/or avoiding harm," which mechanisms may include "tamper resistance" and "authentication," the latter separately defined to mean "[i]dentifying (*e.g.* a ... device ... includ[ing] uniquely identifying." The software contains both the encryption tamper-resistance feature described above, and authentication – programming that creates and uses machine signatures to uniquely identify hardware and thereby prevent unauthorized use of the software. Inasmuch as both of these forms of security are software-based, the Durst reference discloses all the features of a HPE under either definition of that term.

Claim Language	**a central processing unit**

A central processing unit is a standard computer component –in personal computers, this is typically a microprocessor. The Durst Patent discloses a central processing unit. Durst, Fig. 1; 7:26.

Claim Language	**main memory operatively connected to said central processing unit**

The Durst reference discloses a main memory (RAM) connected to the CPU. Durst, Fig. 1; 7:18-20.

Claim Language	**mass storage operatively connected to said central processing unit and said main memory**

The Durst reference discloses mass storage (disk drive) connected to the CPU and main memory. Durst, Fig. 1; 8:15-18 ("... for convenience, the following description is directed to software embodied in the form of a floppy disk, although the specification should be interpreted to include ... other mass storage devices"); 9:3-4 ("Disk drive 116 may take the form of a floppy disk drive or a fixed disk drive, the latter also being referred to as a 'hard' or 'Winchester' disk drive").

Claim Language	said mass storage storing tamper resistant software designed to be loaded into said main memory and executed by said central processing unit, said tamper resistant software comprising

The Durst software is tamper resistant (see discussion of HPE claim element, above). It is, in the standard fashion, loaded from mass storage (*e.g.*, a hard or floppy disk drive) into main memory (*e.g.*, RAM) and executed by the CPU.

4. The Programming Is The Same

The "programming" in the claims at issue has three aspects: "machine check programming," which undertakes the generation and storage of the machine signature based on HPE information; "integrity programming," which activates the machine check programming to re-generate the machine signature and compares the result with the stored signature; and "programming which takes one or more actions" based on the result of the comparison. The Durst Patent discloses all of these.

a. Machine Check Programming

Claim Language	machine check programming which derives information from one or more aspects of said host processing environment, one or more storage locations storing said information

(1) The Meaning of This Element

"Machine check programming" is a module that derives information from one or more aspects of the HPE. The court has defined "derive" to mean "obtain, receive, or arrive at through a process of reasoning or deduction. In the context of computer operations, the 'process of reasoning or deduction' constitutes operations carried out by the computer." Markman Order, at 21. In other words, the computer programming carries out operations on aspects of the computing environment to produce data in some form (the machine signature), which it then stores.

The parties agree that this claim language applies to any derivation of information that represents an attribute of the hardware on which the machine-check programming is running. Throughout its infringement chart, for instance, InterTrust matches this language with the

///

///

following description of an infringing element: "derives from the client computer . . . *hardware ID information*." InterTrust's Amended Disclosures of Asserted Claims and Preliminary Infringement Contentions ("IT's Amended Disclosures"), at 18, 20, 34, 36, 38, 40, 42, 44 (emphasis added). In short, the machine signature may be based on hardware information.

The parties also agree that hardware ID information can be based on any parameter of the physical, material part of the computer, such as "one or more of the CDROM device, disk adapter, disk device, display adapter, first drive serial number, MAC address, processor serial, processor type, RAM size, SCSI adapter, PCMCIA controller, audio adapter, and whether the computer is dockable." IT's Amended Disclosures, at 25. Elsewhere in its chart, InterTrust lists an overlapping but somewhat different set of hardware attributes that could serve as the source of the derived information. Microsoft agrees that any hardware parameters will do.

"Machine check programming" *cannot*, however, refer to the derivation of attributes solely from software files stored on the system. InterTrust has taken inconsistent positions on this point, arguing that even a software module that derives its checkable values entirely from such files can constitute "machine check programming." *See, e.g.*, IT's Amended Disclosure, at 23 (accusing Windows File Protection). InterTrust's inconsistency is immaterial to this motion as Durst clearly teaches deriving information from hardware, which satisfies the requirements of § 102(b) anticipation.

(2) Machine Check Programming in the Durst Reference

The Durst Patent discloses machine-check programming that generates a machine signature from hardware parameters and stores it. The software contains a "measure signature" step, Durst, Fig. 14 (and see generally 26:55-27:31), and "the 'signature' of a computer system is intended to refer to the values of certain characteristics exhibited by that system." Durst, 3:45-47. The characteristics can be of two types: "(a) parameters which are designed specifically into individual computer systems (such as the type of processor, the version of operating software, etc.), and (b) parameters which are defined by particular tolerances in the manufacture of the computer system and its peripherals (*e.g.*, the specific rotating speed of a disk drive, which may

///

vary within a range of design tolerances, etc.)." *Id.*, 3:60-68. Much of Durst's written description explains how to measure particular hardware characteristics in order to create a machine signature, such as the

> identification of the computer system processor, the clock speed of the computer system clock generator, an identification of the computer system ROM, the wait time, or wait cycles, assigned to the computer system processor for accessing a RAM, the rotary speed of a computer system disk drive, the access speed of that disk drive and the sector interleave value of that disk drive.

Id., 3:50-57; col. 11 – col. 25 (detailed description of measuring techniques). However, "[t]he invention is not intended to be limited solely to these examples; and other characteristics which can be used to distinguish one computer system from another are contemplated." *Id.*, 3:57-60. The signature is "determined in accordance with the subroutines" that extract these various hardware measurements, as described in columns 11-25. Durst, 25:58-60.

The Durst reference also discloses "one or more storage locations storing said information": "After the signature of the computer system has been measured, it is recorded, or stored, in the software integrated with the applications program." *Id.*, 26:14-16; *also* 27:11-13.

b. Integrity Programming

Claim Language	**integrity programming which** **causes said machine check programming to derive said information** **compares said information to information previously stored in said one or more storage locations, and** **generates an indication based on the result of said comparison**

(1) The Meaning of This Element

The integrity programming activates the machine check programming, causing it to derive information based on HPE parameters in the same manner as discussed above, to compare the result to the previously stored result, and to generate an indication reflecting the outcome of that comparison.

An aside is needed regarding the phrase "said information." This language is slightly confusing in that it might be taken to mean that the *results* of the derivation of information must be the same as the previously stored information. Yet the purported invention's functionality depends on comparing the latter result with the machine signature previously stored

to determine if the two are different. Thus, "said information" must mean information derived in the same manner by the same programming, but which may lead to a different value each time it is run. This construction of the term is supported by the specification, '900 Patent, 239:4-25, and by InterTrust's own infringement chart.[3] IT's Amended Disclosures, at 28.

(2) Integrity Programming in the Durst Reference

Just as in the '900 Patent claims, the Durst reference discloses programming which causes the machine signature to be derived, compares it with the stored signature, and produces an indication based on the result. On this point, the language of the Durst Patent is such that a comparison chart is the most efficient way to demonstrate the correspondence between the claim language and the Durst reference:

integrity programming which	"The copy protection procedure inquires initially at 1402 if a signature has been stored previously on the floppy disk. If this inquiry is answered in the affirmative," (26:59-62; Fig. 14)
causes said machine check programming to derive said information,	"then the signature of the computer system with which the applications program is intended to be run is measured." (26:62-64)
compares said information to information previously stored in said one or more storage locations, and	"If the measured signature is the same as the previously determined and stored signature, inquiry 1412 is answered in the affirmative and the applications program is executed, as represented by instruction 1408. However, if inquiry 1412 is answered in the negative, an error message is displayed, thereby indicating that an attempt has been made to run the applications program on an unauthorized computer system." (26:64-27:3)
generates an indication based on the result of said comparison; and	

///

///

///

///

[3] Microsoft rejects InterTrust's infringement assertions as to its products and cites InterTrust's infringement position only to show that the parties are in agreement on the relationship between the two different hardware checks that the software performs.

c. **Programming That Undertakes an Action Based on the Comparison Result**

Claim Language	programming which takes one or more actions based on the state of said indication
Claim 155 only	*said one or more actions including at least temporarily halting further processing.*
Claim 156 only	*said one or more actions including at least temporarily disabling certain functions.*
Claim 157 only	*said one or more actions including displaying a message to the user.*

The action the software takes upon discovering a discrepancy between the previous and the current machine signature is the only respect in which claims 155, 156 and 157 differ from one another. The Durst reference discloses a response to an attempt at unauthorized use of the software that satisfies each of these three different claim elements:[4] "[I]f inquiry 1412 [the check of whether the present and stored signatures match] is answered in the negative, an error message is displayed, thereby indicating that an attempt has been made to run the applications program on an unauthorized computer system. It is appreciated that, under this condition, the applications program cannot be executed." Durst, 26:68-27:5. This clearly meets the limitations of displaying a message to the user and disabling certain functions, respectively.

Regarding "at least temporarily halting processing," the Durst Patent discloses that the consequence of a negative comparison of machine signatures is to halt processing of the protected software. Durst, Figs. 13B, 14, 15; col. 26:68-27:5.

///

///

///

///

///

///

///

[4] Microsoft notes that the claim language, read plainly, actually requires that the programming take one or more actions regardless of the outcome of the comparison: "programming *which takes* one or more actions based on the state of said comparison."

IV. CONCLUSION

Because the Durst Patent disclosure has each and every element of the challenged claims Microsoft respectfully requests that the Court declare claims 155, 156 and 157 of U.S. Patent No. 5,892,900 to be invalid as anticipated by a prior patent, pursuant to 35 U.S.C. § 102(b).

Dated: February 23, 2004

BY: _____

WILLIAM L. ANTHONY
ERIC L. WESENBERG
HEIDI L. KEEFE
KENNETH J. HALPERN
SAM O'ROURKE
ORRICK HERRINGTON & SUTCLIFFE, LLP
1000 Marsh Road
Menlo Park, CA 94025
Telephone: (650) 614-7400

STEVEN ALEXANDER
KRISTIN L. CLEVELAND
JAMES E. GERINGER
JOHN D. VANDENBERG
KLARQUIST SPARKMAN, LLP
One World Trade Center, Suite 1600
121 S.W. Salmon Street
Portland, OR 97204
Telephone: (503) 226-7391

Attorneys for Defendant and Counterclaimant
MICROSOFT CORPORATION

Of Counsel:

T. Andrew Culbert, Esq.
One Microsoft Way
Building 8
Redmond, WA 98052-6399
Phone: (425) 882-8080